Carlos Rivera and Payman Arabshahi

Department of Electrical Engineering
University of Washington FT-10
Seattle, WA 98195 USA

Fax: (206) 543-3842
e-mail: payman@u.washington.edu

Abstract— In the present paper, we propose a fuzzy vector quantization algorithm based on the the well known LBG algorithm [8], which has been successfully applied in areas such as speech and image processing. The principal goal of our algorithm is to derive a fuzzy partition as described in [7].

In order to achieve this, we define membership functions derived from probabilistic principles and depending on unknown *a priori* probabilities. The maximum entropy principle is used to determine these *a priori* probabilities. Examples for cases which result in the choice of uniform and Gaussian densities are presented.

Vector Quantization

Vector quantization (VQ) [1] is a technique that has been successfully applied in some problems of data compression and pattern recognition. It is based on the well known result of Shannon's rate-distortion theory which states that the performance of a data compression system is improved by coding blocks of data; with larger blocks giving better performance. There is a limit for this improvement however, which is related to the entropy of the source and the distortion that one is allowed to have.

For instance in the case of a continuous Gaussian source, it is not difficult to see that regardless of the length of the block, we will never obtain a perfect representation. This is because a real number needs an infinite number of bits for representation. Thus for such a source we will never be able to obtain zero distortion. On the other hand, if some average distortion d_0 is permitted, it is possible to approximate the source arbitrarily close to d_0.

With this principle in mind, several algorithms have been proposed for vector quantization. In general, their goal is to obtain a set of points ("codes") that for a given distortion measure minimizes the total error in the representation of the source.

Two questions immediately arise: What distortion measure should be used? And, how many points are necessary to achieve this goal? To answer the first question, it is necessary to look at the particular problem at hand. For instance in the area of speech processing, one distortion measure that has been used is the Itakura-Saito measure [2].

The answer to the second question depends on the amount of total distortion that we are allowed to have. In some cases it is known that for some algorithms, initial conditions will affect the number of points.

Over the years, fuzzy set theoretic concepts, first developed by Zadeh [3], have found applications in the area of pattern recognition [4]. Recently, a classified fuzzy vector quantization technique has been proposed and applied to the problem of image compression [5]. In this paper we propose another algorithm for fuzzy vector quantization, and consider its application to a clustering problem.

To motivate the discussion of our proposed algorithm, we present a brief overview of three other widely used vector quantization algorithms.

This is one of the more widely used of the VQ algorithms [6]:

Given a training sequence formed by a large collection of points - which we assume represents the source - and an initial set of N points, C_i^0, $i = 1, \ldots, N$, that we call centroids,

i) Find the regions that these N centroids define.
ii) Find the new centroids of these regions, C_i^1, $i = 1, \ldots, N$.
iii) Calculate the total distortion, d_T.
iv) Calculate ε_T, the difference between C_i^0 and C_i^1.
v) Stop if ε_T or d_T is below a predetermined threshold.
Otherwise set $C_i^0 = C_i^1$, $i = 1, \ldots, N$, and go to i.

■ Fuzzy c-means algorithm

This algorithm [7] constructs fuzzy partitions of data space, and assigns N membership functions to the N partitions.

Given a training sequence formed by a large collection of points, which accurately represent the space

i) Initialize the partition matrix $U^{(0)}$.
ii) Calculate the fuzzy cluster centers $\{v_i\}$.
iii) Update the partition matrix, $U^{(1)}$.
iv) Calculate ε_T, the difference between $U^{(0)}$ and $U^{(1)}$.
v) Stop if ε_T is below a predetermined threshold.
Otherwise set $U^{(0)} = U^{(1)}$ and go to ii.

■ LBG Algorithm

One very important variation of the *k-means* algorithm and one that we will use for our study is the LBG algorithm [8], which we reproduce here:

Given a training sequence formed by a large collection of points, which we assume represents the source:

i) Find the general centroid, C_i^1, $K = 1$, $i = K$
ii) Split every C_i^1, $i = 1, .., K$, in two new centroids,
C_i^0, $i = 1, \ldots, 2K$; set $K = 2K$
iii) Find the K regions that these centroids define.
iv) Find the total distortion d_T.
v) Find the centroids of these new regions, C_i^1, $i = 1, \ldots, K$.
vi) Find ε_T, the difference between C_i^0 and C_i^1,
vii) If d_T or ε_T is not below a predetermined threshold,
set $C_i^0 = C_i^1$, $i = 1, \ldots, N$. Then go to iii.
viii) If $K = N$, end, otherwise go to ii.

of $i = 1, 2, 4, ..., N$, i.e. it give us the $1, 2, 4, ..., N$, centroids that are at least locally optimum.

Fuzzy LBG algorithm

Using the idea of the LBG algorithm, we propose the following scheme to obtain a fuzzy partition,

i) Find the general centroid, $K = 1$, C_i^1, $i = K$. Find the membership function values.
ii) Split every C_i^1, $i = 1, .., K$, into two new centroids, C_i^0, $i = 1, ..., 2K$; set $K = 2K$
iii) Find the K regions that these centroids define. Calculate the membership function for each region.
iv) Find the total distortion d_T.
v) Find the centroids of these new regions, C_i^1, $i = 1, ..., K$.
vi) Find ε_T, the difference between C_i^0 and C_i^1.
vii) If d_T or ε_T is not below a predetermined threshold, set $C_i^0 = C_i^1$. Then go to iii.
viii) If $K = N$, End. Otherwise go to ii.

The most important issue in this algorithm is how to define the membership function for each region. One idea that comes very naturally is using the probability that given a point x_j it belongs to the partition A_i, i.e. $p(A_i|x_j)$. Using Bayes formula we have:

$$p(A_i|x_j) = \frac{p(x_j|A_i)p(A_i)}{\sum_k p(x_j|A_k)p(A_k)}$$

or, in the continuous case [9]

$$p(A_i|x) = \frac{f(x|A_i)p(A_i)}{\sum_k f(x|A_k)p(A_k)}.$$

In order to determine the values of $p(A_i)$ we use the simplest possible estimator, i.e. $p(A_i)$ = number of points in region A_i/ total number of points.

To estimate the values of $p(x_j|A_i)$, or the densities $f(x|A_i)$ for the continuous case, a different approach has to be taken. In this paper, we use the maximum entropy approach [9], for estimation of the density functions $f(x|A_i)$. In one case, we assume that only the region of support in which the vector x can occur is known and that this region is finite. This results in a uniform density function. The second example corresponds to the case of known variance, resulting in a Gaussian density function.

Results

In order to test the algorithm, we applied it to the example presented in [10] and analyzed in [7]. The data set consists of 15 points in $\mathbf{R^2}$, as shown in Fig. 1. Data points (2,2), (3,2), and (4,2) form a bridge or neck between the wings of the butterfly. A possible interpretation of this pattern is that the points in the wings are from two distinct classes and that the points in the bridge are noise.

The objective is to find a fuzzy partition of this space and gain some insights into the selection of the *a priori* $f(x|A_i)$ densities.

As it can be seen from Fig. 1, the data space can be intuitively separated into two regions: All the points to the left of the point (3,2) belong to region A_1; while all the points to its right belong to region A_2. The point (3,2) itself is difficult to assign, due to the symmetry of the data. We expect that any algorithm that partitions the space will give a result similar to that obtained by intuition.

Eucledian distance measure. As expected the point (3,2) is assigned as the general centroid. Figure 3 shows the two centroids when the space is divided into two regions. It can be observed that because of the characteristics of the algorithm, point (3,2) is assigned to region A_1. As expected the algorithm has separated the space into two regions that are very similar to those that we expected.

Figure 4 and Table 1 present the results when the fuzzy vector quantizer is applied to the data. An assumption is made that we only know the finite region of support in which the noise occurs. The maximum entropy principle together with the previous assumption results in the choice of the uniform density for $f(x|A_i)$. Table 1 shows the resulting fuzzy partition. Note the similarity to crisp (non-fuzzy) partitioning.

Finally, Figure 4 and Table 2 present the results when the variance of the noise is assumed known; in this case the resulting $f(x|A_i)$ is Gaussian. From Table 2 we can observe that points that are close to the border of the regions, are assigned values close to 0.5, thus reflecting the ambiguity in assigning such points to one or the other region.

References

[1] A. Gersho and R.M. Gray, *Vector quantization and signal compression.* Boston: Kluwer Academic, 1992.

[2] R.M. Gray, A. Buzo, A.H. Gray Jr., Y. Matsuyama, "Distortion Measures for Speech Processing," *IEEE Trans. Acoustics, Speech, Signal Processing,* Vol. 28, No. 4, 1980.

[3] L.A. Zadeh, "Fuzzy sets," *Information & Control,* Vol. 8, pp. 338–353, 1965.

[4] J.C. Bezdek and S.K. Pal, eds., *Fuzzy Models For Pattern Recognition.* New York: IEEE Press, 1992.

[5] F. Marqués and C.-C. Jay Kuo, "Classified vector quantization using fuzzy theory," *Proc. First IEEE Int. Conf. Fuzzy Systems, FUZZ-IEEE '92,* San Diego, CA, 1992.

[6] J.T Tou and R.C. Gonzalez, *Pattern recognition principles.* Reading : Addison-Wesley, 1974.

[7] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms.* New York: Plenum Press, 1981.

[8] Y. Linde, A. Buzo, R. M. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Trans. Communications,* Vol. 28, No. 1, 1980.

[9] A. Papoulis, *Probability, Random Variables, and Stochastic Processes.* New York: McGraw-Hill, 1984.

[10] E. Ruspini, "Numerical Methods for Fuzzy Clustering," *Information Sciences,* Vol. 2, pp. 319–350, 1970.

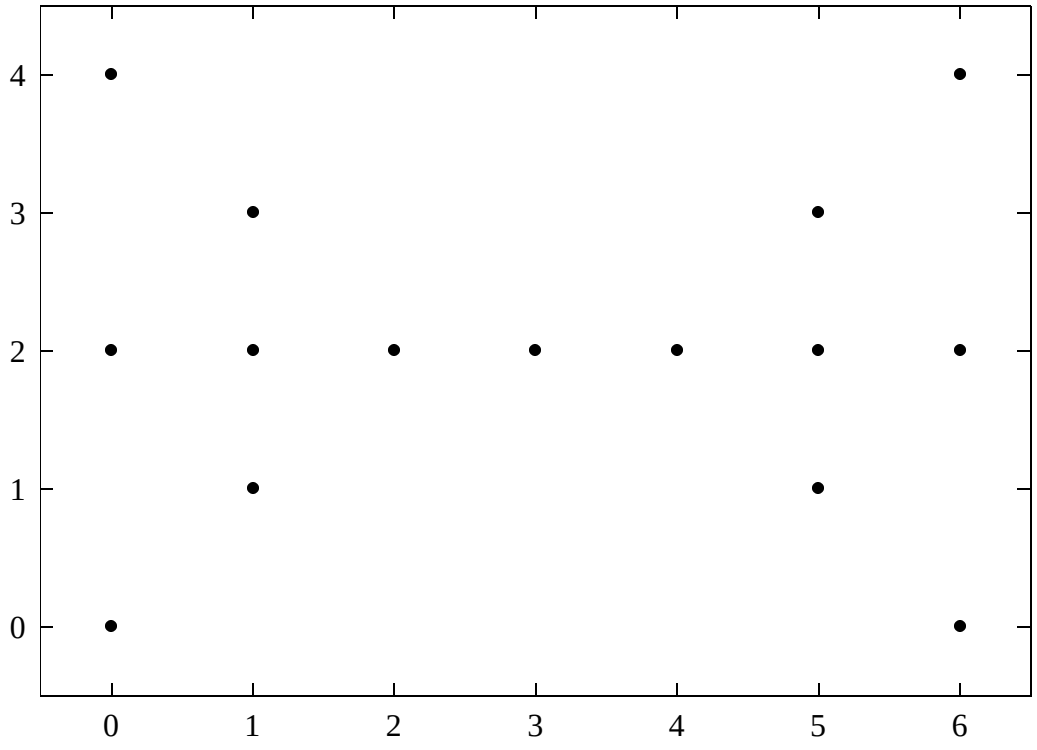

Figure 1: Data used in the examples, (from [10]).

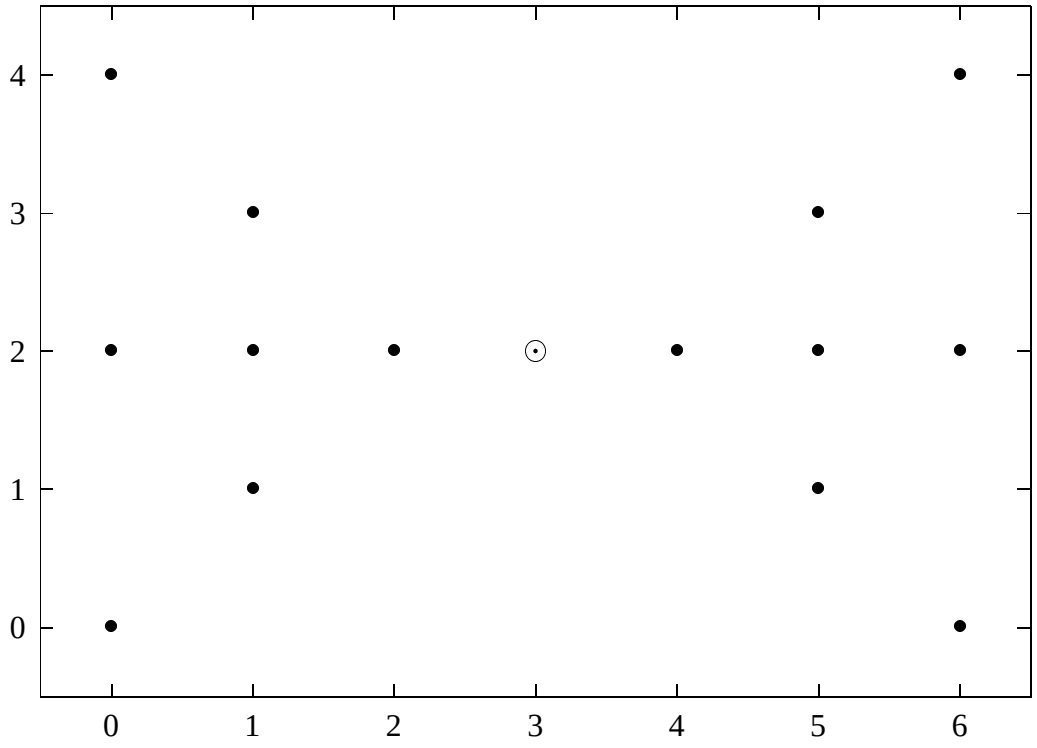

Figure 2: First step of the LBG algorithm; the point ⊙ represents the general centroid.

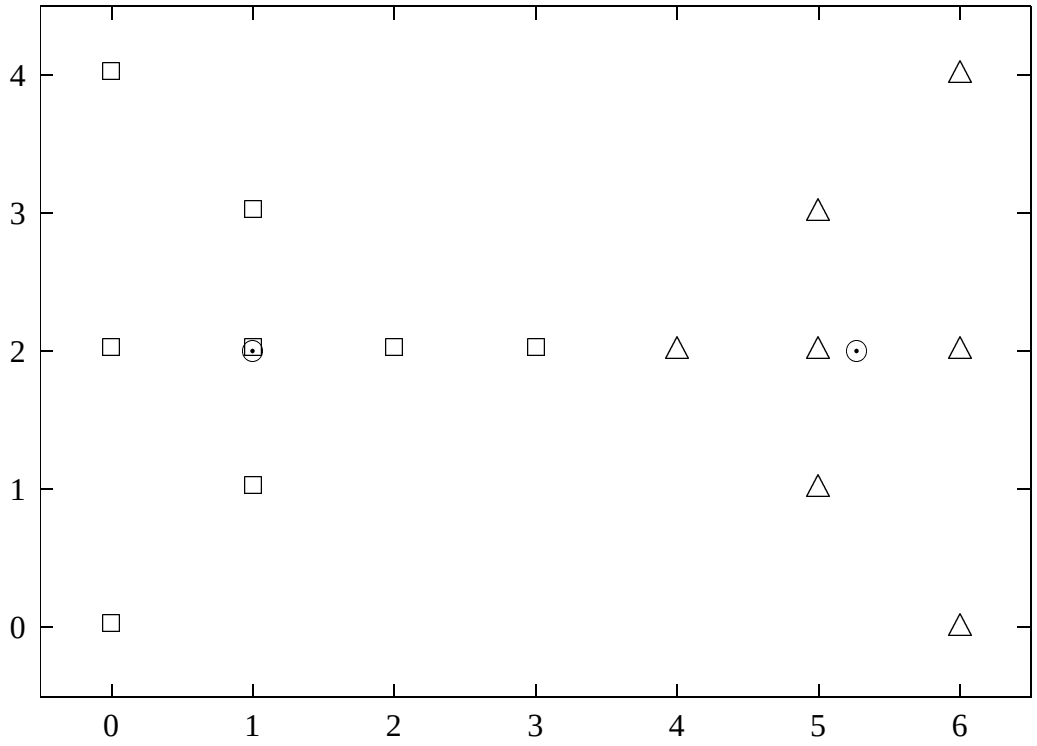

Figure 3: Final result of the LBG algorithm when the space is divided into two regions: The □ points belong to region A_1, and the △ points belong to region A_2. The centroids of each region are represented by ⊙.

x	$\mu_x(A_1)$	$\mu_x(A_2)$
(0,0)	1	0
(0,2)	1	0
(0,4)	1	0
(1,1)	1	0
(1,2)	1	0
(1,3)	1	0
(2,2)	1	0
(3,2)	8/15	7/15
(4,2)	8/15	7/15
(5,1)	0	1
(5,2)	0	1
(5,3)	0	1
(6,0)	0	1
(6,2)	0	1
(6,4)	0	1

Table 1: Membership function values when the resulting pdf $f(x|A_i)$ is uniform.

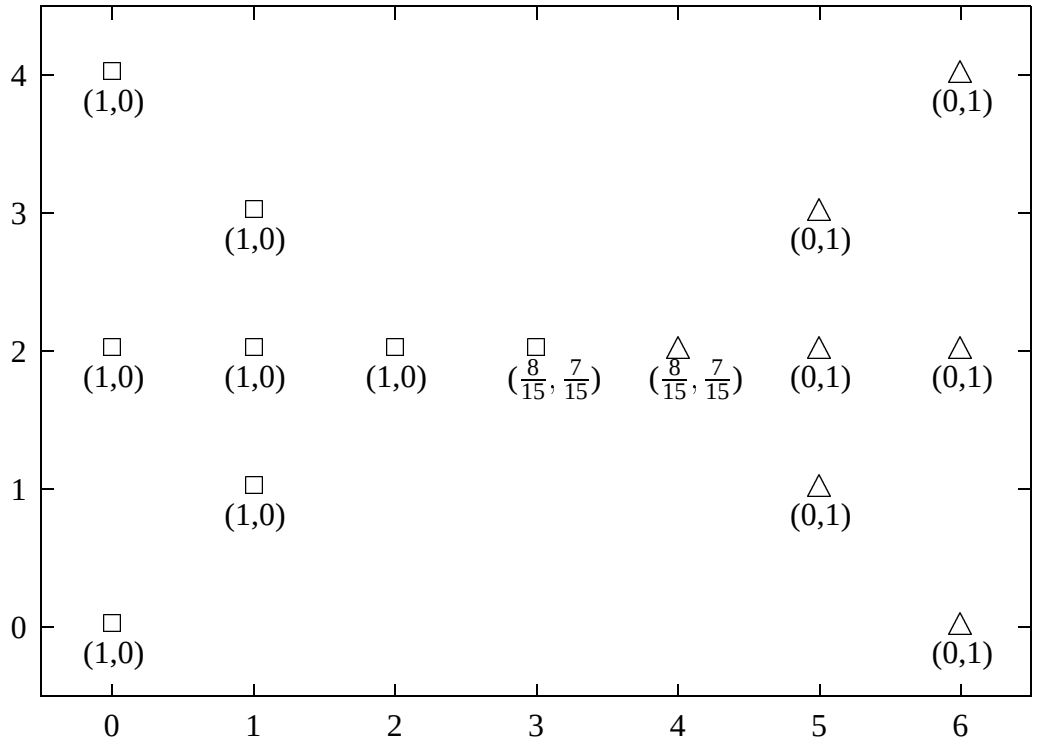

Figure 4: Fuzzy partition when the resulting $f(x|A_i)$ is uniform. The numbers inside the parenthesis represent the values of the membership functions, i.e. $(\mu_x(A_1), \mu_x(A_2))$. Note the similarity with crisp partitioning.

x	$\mu_x(A_1)$	$\mu_x(A_2)$
(0,0)	0.97071	2.9287E-02
(0,2)	0.97071	2.9287E-02
(0,4)	0.97071	2.9287E-02
(1,1)	0.91904	8.0954E-02
(1,2)	0.91904	8.0954E-02
(1,3)	0.91904	8.0954E-02
(2,2)	0.79543	0.20456
(3,2)	0.57116	0.42883
(4,2)	0.31327	0.68672
(5,1)	0.13513	0.86486
(5,2)	0.13513	0.86486
(5,3)	0.13513	0.86486
(6,0)	5.0801E-02	0.94919
(6,2)	5.0801E-02	0.94919
(6,4)	5.0801E-02	0.94919

Table 2: Membership function values when the resulting pdf $f(x|A_i)$ is Gaussian. In this case the membership value of the points which are close to the border between regions reflect the ambiguity in assignment.

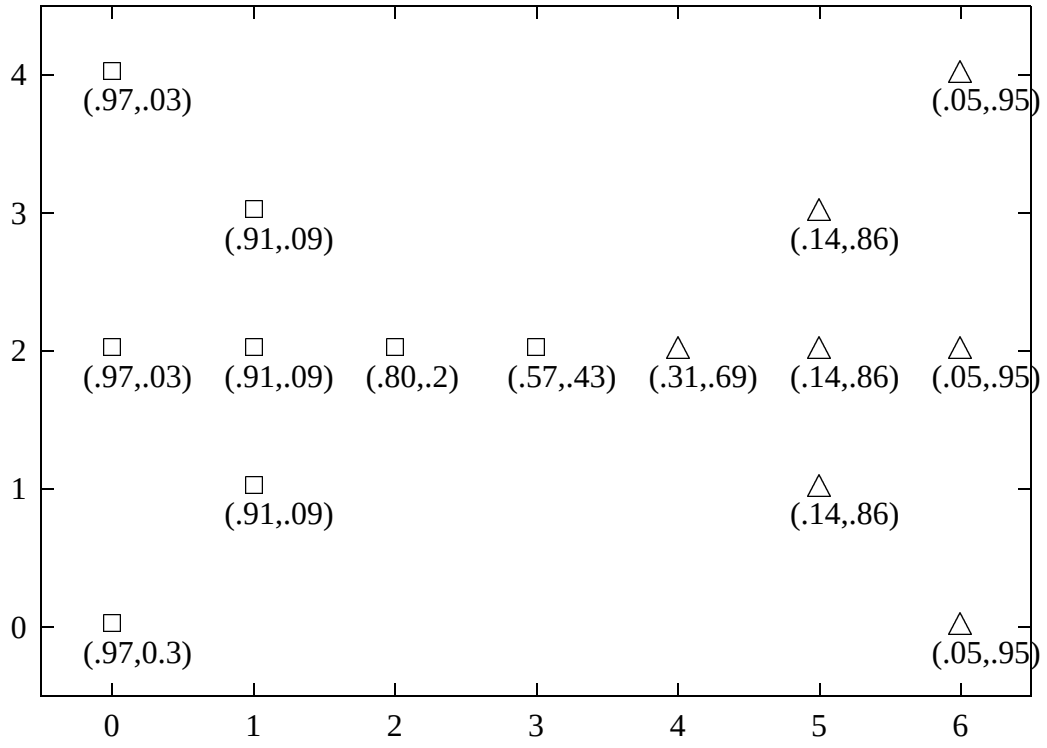

Figure 5: Fuzzy partition when the resulting $f(x|A_i)$ is Gaussian. The numbers inside the parenthesis represent the values of the membership functions, i.e. $(\mu_x(A_1), \mu_x(A_2))$.